iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** X **Month of publication:** October 2022

DOI: https://doi.org/10.22214/ijraset.2022.47214

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue X Oct 2022- Available at www.ijraset.com

An Intelligent Approach for Adaptive Liquid Level Control System Using Machine Learning

Bhagirathi Bai V[1], Hanume Gowda Chandrakanth[2]
[1]*Department of Mechatronics Engineering, Acharya Institute of Technology, India*
[2]*Department of Electronics and Communication, Sambhram Institute of Technology, India*

Abstract: Liquid level control is a mechanism that monitors, compares, and regulates the level of liquids within a process to a set value. Level measurement determines the position of the level relative to the top or bottom of the process fluid storage tank or silo. Level control and measurement are essential to assuring the safety and profitability of industrial processes. In this project, we are Implementing an intelligent mechanism using machine learning to predict the level of liquid in a reservoir based on the time and motor speed thus avoiding errors and keeping the process is ongoing.
Keywords: Adaptive Liquid level control, Machine learning, Intelligent system, reduced error , improved efficiency

I. INTRODUCTION

A liquid Level Control System is a system specifically designed to control the level of fluid in tanks. The main aim possessed by these systems is to control the rate at which the pump delivers fluid to the tank so it can reach the desired level inside the tank.
The purpose of the liquid level system is to maintain a specific level of fluid inside the tank. The liquid level control systems find major applications in industrial processes. The controller present in the system generates the control signal that is converted into the desired signal by the actuator and is fed to the plant to perform the desired action.

A. Errors Occur in the Existing Plant

1) *Case 1:* Flow monitoring
2) *Case 2:* Orientation of the nozzle- if the nozzle is m misaligned, there would be a loss.
3) *Case 3:* If there is a blockage in the pipe or nozzle it is very difficult to find the area of blockage or sometimes it is unknown whether the tank is empty or there is a blockage in the pipe due to which the flow of fluid is stopped.
4) *Case 4:* When the tank gets empty the containers are left unfilled with the liquid and if both the tanks are full the liquid overflows out of the tank.
5) *Case 5:* When the motor has to be running at a speed of 20 rpm but rotates at 30 rpm then the outflow is more and the tank level will increase.
6) *Case 6:* Based on the product demand, the quantity of liquid level in the tank has to be ensured.

II. LITERATURE SURVEY

The water level in the tank[1] is controlled according to the set point given by the user,this approach was chosen mainly because fuzzy logic is more accurate and stable for a wide range of set points. The water levels were reached accurately and the time taken to reach the set point depends proportionately upon the value of the error. But the Level in the tank was checked manually. Error Measured Was 2.5. Efficiency was for Different Set Points and Achieved 95.3% 8 Level Set Points Were Used.
Dongsheng Wanga,*, Yan Wang [2] proposes a waterworks optimization control system based on machine learning, through the historical data of the plant water treatment process. offered a new way for optimal control of water, and at the same time confirmed the effectiveness of machine learning in the application of water energy consumption prediction and configuration optimization direction. Second, this paper designs an optimization control system for water plant optimization, consisting of a concise and convenient control interface for the above subsystems, and combines the theory and the use of the water plant optimization control system.This author does not mention the auto correcting of the speed. The efficiency achieved was only 60.9 % .
Kalidasan, J. Ben Ajai Raja proposed an Automatic Bottle Filling Machine [3] where the Motor Is Used to Rotate the Bottle base plate, and Sensors were Used to identify the Position of the Base Plate Rotation. Then the Solenoid Valves are used To open the Valve Of The liquid flow to the bottle for a Particular Time, in this work the quantity of flow of liquid in the tank is not considered, it does not have a feedback loop. And No monitoring of level in the tank and no automatic liquid filling is considered

©IJRASET: All Rights are Reserved

Filling Machine which can fill different sizes of containers based on the height[4]. The filling of liquid is done based on the threshold set in the controller. In this work Identifying the blockages and errors in flow pipes are not considered. and there is no monitoring of the liquid level in the tank. And no mechanism for auto refilling of the tank was deduced.

Md. Liton Ahmedl proposed a n automated bottle filling system using a PLC based controller [5]. The filling and the movement of the bottle along the plant is controlled by the afore mentioned PLC. The PLC controls a conveyor on which the place unfilled bottle move. The position of the bottles are detected by an infrared sensor and a dc pump is used to control the flow of water, hence enabling functionality at night. Here the proposed method doesnot take into consideration the quantity of liquid filled or the monitoring of the level of liquid in the tank. Also it lacks the implementation of a feedback loop and a prediction model to estimate the speed of the motor.

The idea and development of an intelligent tank with a water level sensor [7] is conceptualized wherein the water proportion can be determined by the emptiness and fullness of the tank using artificial intelligence. The proposed system claims the reduction in water wastage but lacks proposing an auto correcting closed loop system. The system also lacks real world efficiency.

The proposed system of ultrasonic transducer based water level sensor using pump switching [8] addresses the problems of untimely response and frequent breakdown of contact sensors due to corrosion and coating from the water medium. Yet it fails to address scenarios where the pump would run unnecessarily when there is no water at the source and overheating of the pump.

Liquid level monitoring using FBG sensor array [9] was also proposed 9 FBG(Fiber Bragg Grating) temperature sensors in a tank of height 100 cm and width 30 cm. The work proposes a level estimation in two steps where level detection is done using classification models and level estimation is done using regression models. The level detection is carried out by calculating under which FBG the liquid level is. The proposed method has error of 3.56 cm to 6.28 cm depending on the machine learning model. It only has a claimed efficiency of 89.50%.

As per our proposed solution, the control of a dc pump using machine learning is a effective and feasible solution. Considering the study done on the comparison between PID control and Neural network control over a dc motor [10], the concept of using newer trained models and logics over conventional PIDs is proven better. Similar work has been done to develop a mechatronic model that monitors the water levels and controls the inflow rate using fuzzy logic [11]. The simulation showed an average 0.345 error with an efficiency on only 75.50%.

The effectiveness of the afore mentioned approach is further emphasized the comparison with fuzzy logics[12], where in the intrustrial controller like PID had an estimated steady state error of 3.56 seconds.

The implantation of IOT for Industrial Monitoring and Control [13] was also proposed. The proposed method relied on the use of a reliable TCP/IP protocol in conjunction with GPRS enabled communication. This was done using a GSM and DTMF module. The proposed system was based on a conventional monitoring system with no auto control or implementation of any prediction models to increase efficiency and effectiveness.

III. OBJECTIVE

The proposed work is to develop a smart intelligent adaptive system which auto controls the level of liquid in the tank based on the data learn by prediction models to increase efficiency and effectiveness. The proposed solution is as follows.

IV. PROPOSED SOLUTION

As we can observe from the above-stated literature survey some plants require human interference in the work cell to avoid errors like checking tank level, blockages, flow, leaks, auto refilling, etc. and the systems are not intelligent enough to reduce the error and get the maximum efficiency.and also, these existing system does not consist of any monitoring display to show the parameter values. To overcome these disadvantages, we introduce a hybrid Machine Learning system that can monitor and detects an error in the process and auto-correct it intelligently.

The process is integrated with Machine Learning algorithm[15] which was implemented to control the motor speed for better accuracy and less error,which keeps a record of old input commands and compares them with the given command with the help of a database and suggests the best action to be carried out during the period. Even in case, the command is not given it uses the existing commands from the machine-learned database and executes to solve the particular error.

This system consists of two modes, semi-automatic mode and an automatic mode wherein semi-automatic mode the human interference is involved and the input is given to the system through speech commands[14] or GUI. in automatic mode, the system analyses the problem and compares it with the database to solves the situation accordingly.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

V. CIRCUIT DIAGRAM AND FLOW CHART

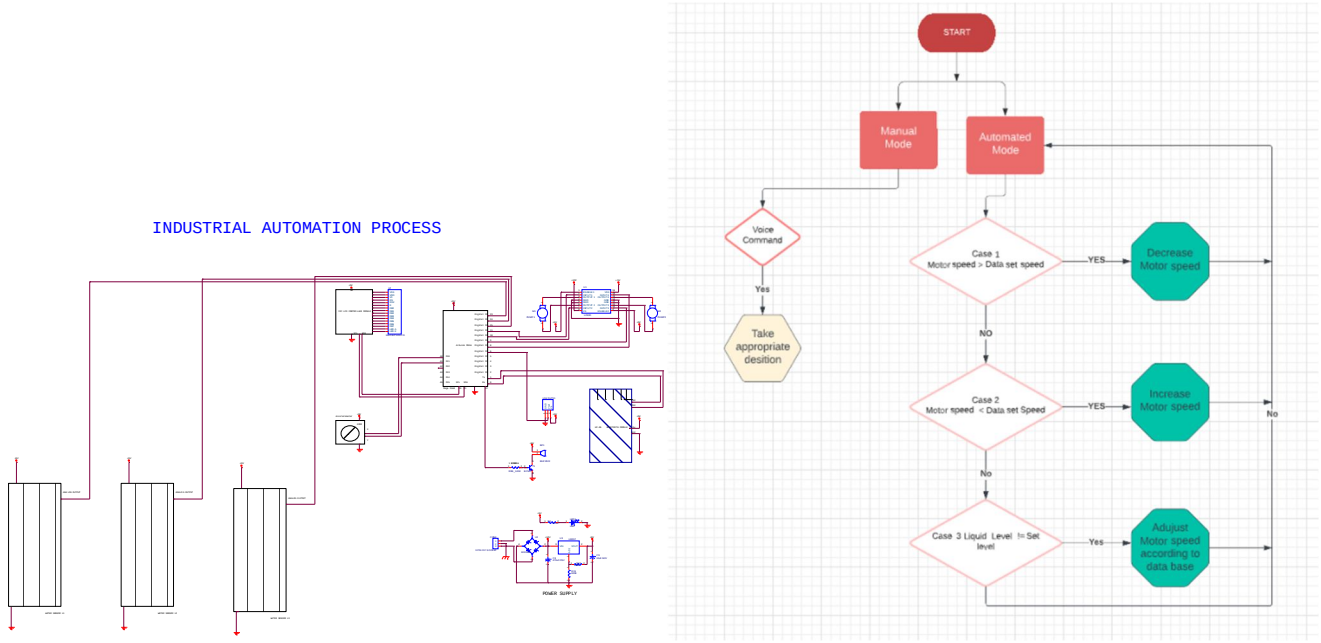


Fig 1: Circuit diagram and flowchart

1) The Machine Learning algorithm [15]compares with the previous data set at that particular time and corrects the rpm according to it and reduces the error.

2) A system in which the Machine Learning algorithm[15] controls the plant with comparison of the previous database where the algorithms is written in such a way to reduce the error and maximize the efficiency of the production. Where the Machine learning code consist of speech recognition where it filters the unregistered speech frequency through database. The micro-controller displays all the values of the sensors in the display attached to it. The Machine learning system monitors the flow, liquid output and level of the tank.

3) Where to overcome this scenario data sets which consists of day, time, level of water and motors speed are being recorded.

4) Then the recorded data is processed by the ML algorithm[15] and gives the prediction of fluid required to filled the next day.

5) The ML code compares the data of real time output of the motor speed and the reference speed set from the data set by the trained model to detect the error.

6) The ML code adjusts the motor speed to minimize the error according to the reference speed from the data set.

VI. PLANT DESCRIPTION & DIAGRAM

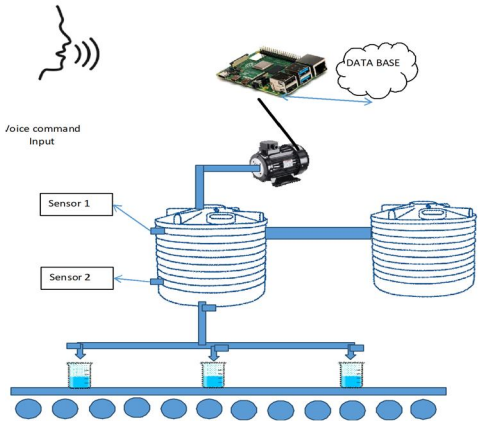


Fig 2: Block digram of plant with sensors and database

let's assume a plant where the tanks are filled with liquid and they flow through the pipe goes to the nozzle where the containers come through the conveyor belt and the nozzle valve opens and fills the containers with the liquid, Where to be precise the primary tank is filled with the liquid with the help of a pump and when tank A is empty or goes below the threshold the liquid is pumped to tank A from the secondary tank.

1) The input command is given through voice or manually[15] is executed by the controller.
2) Where the controller is connected to the database and checks or compares the input commands parameters with the previous old commands parameters which are stored in the database.
3) Cu(control unit) will decode the tag, and take appropriate decisions on pump automation.
4) The control unit is connected with different types of sensors and actuators to control the error of the operation and increase the efficiency of the production through uninterrupted services.
5) There are three sensors connected to the tank. L3 (on top) L2(in middle) , L1(bottom).
6) As shown in Fig 2, If L1, L2, L3 =1(negative logic); no sensor data detected= tank is empty and the process will be stopped.
7) If the L1, L2=1; L3=0, then the pump speed in the primary will be reduced to 20%.
8) If the L1=1; L2=0; L3=0, then the pump speed is increased by 50%.
9) If the L1=0; L2=0; L3=0, then the pump speed is increased by 100%.
10) If the Nozzle position is 20<=x<=200, then it indicates a nozzle error and a message will be sent to the control unit for manual verification as shown in Fig 3
11) The entire process is instantaneously controlled by the fusion of sensor data.



Fig 3: Database showing Nozzle Misalignment,Nozzle position, level sensor, and motor speed.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

VII. RESULT

1) A smart liquid filling machine that adjusts the motor speed based on time.it can also auto-adjust the motor speed in case of any change in the liquid level.
2) Auto-adjust system works on RF(random forest**).** this model is capable of predicting the water level based on the time(run time of system in minutes) and the motor speed
3) The dataset used to train this model consists of the run time of the system in minutes, the motor speed in %, and the water level.
4) The data can be represented in a scatter plot as shown below

Predicting the speed of the motor takes place with three different cases explained below as following

A. Case 1:Interval of 30 min

In case 1 the motor speed and liquid levels are measured at an interval of 30 min as shown in the image below. the results are shown in the below image. we got are accuracy is 0.986 And the mean error is 0.009234 which is the highest accuracy among the other 2 cases.

A	B	C
minutes	motor_spee	level
0	100	3
30	30	1
60	10	1
90	100	3
120	50	2
150	80	3
180	100	3
210	40	2
240	0	1
270	70	3
300	50	2
330	70	3
360	30	1
390	0	1
420	80	3
450	70	3
480	40	2
510	50	2
540	60	2
570	50	2
600	30	1
630	0	1
660	75	3
690	70	3
720	30	1
750	50	2
780	50	2

Fig 4: datasets used to train the model

```
In [7]: customcmap = ListedColormap(["crimson", "mediumblue", "darkmagenta"])
        fig, ax = plt.subplots(figsize=(8, 6))
        plt.scatter(x=df['time'], y=df['motor_speed'], s=150,
                    c=df['level'].astype('category'),
                    cmap = customcmap)
        ax.set_xlabel(r'time', fontsize=14)
        ax.set_ylabel(r'motor_speed', fontsize=14)
        plt.xticks(fontsize=12)
        plt.yticks(fontsize=12)
        plt.show()
```

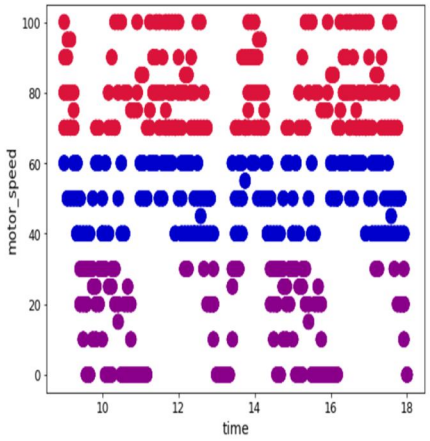


Fig 5: Scatter plot of the Data set

The RF model has the input as the minutes and motor speed and the output as the level

- First, we have to split the data set into train and test data. To do it we have to import the function test train split. this function divides the dataset into train and test in a ratio of 7.5:2.5 as shown in Fig 6.

```
In [14]: from sklearn.model_selection import train_test_split

In [56]: X = motor.drop('level',axis=1)
         y = motor['level']
         X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=100)
```

Fig 6: Training and testing of Data set

- After splitting the data into train and test we have to import RandomForestClassifier and use this function on the training datasets to create an Rf mode as shown in Fig7

```
In [57]: from sklearn.ensemble import RandomForestClassifier

In [58]: rfc = RandomForestClassifier(n_estimators=600)

In [59]: rfc.fit(X_train,y_train)

Out[59]: RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                                criterion='gini', max_depth=None, max_features='auto',
                                max_leaf_nodes=None, max_samples=None,
                                min_impurity_decrease=0.0, min_impurity_split=None,
                                min_samples_leaf=1, min_samples_split=2,
                                min_weight_fraction_leaf=0.0, n_estimators=600,
                                n_jobs=None, oob_score=False, random_state=None,
                                verbose=0, warm_start=False)
```

Fig 7: Classifier

- Now we test our RF model by applying it to the test data and verifying the working and accuracy of the model using a confusion matrix classification report as shown in Fig 8. The accuracy measured is 98.6%

```
In [60]: predictions = dtree.predict(X_test)

In [61]: from sklearn.metrics import classification_report,confusion_matrix

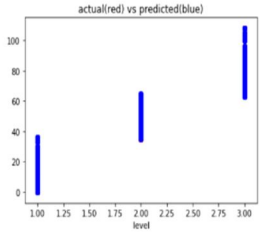
In [62]: print(classification_report(y_test,predictions))
```

	precision	recall	f1-score	support
1	0.986	0.995	0.993	207
2	0.992	0.988	0.996	218
3	0.998	0.996	0.989	225
accuracy			0.986	650
macro avg	0.99	0.99	0.99	650
weighted avg	0.98	0.98	0.98	650

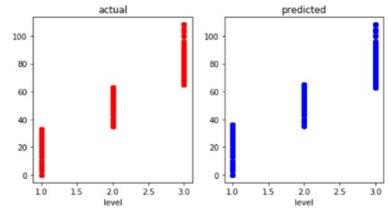
Fig8: Confusion matrix, accuracy

- A scatter Graph is a plot between the actual and predicted water level as shown in Fig 9.

```
In [20]: plt.scatter(y_test,motor,color = 'red')
         plt.scatter(y_pr,motor,color = 'blue')
         plt.title('actual(red) vs predicted(blue)')
         plt.xlabel('level')
         plt.show()
```

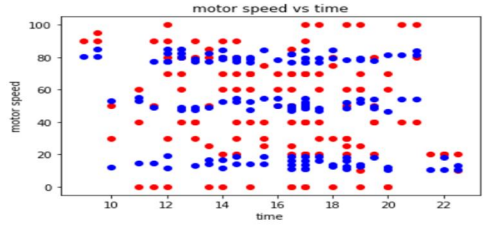



```
In [21]: fig, ax= plt.subplots(figsize=(8,4),ncols=2)
         ax[0].scatter(y_test,motor,color = 'red')
         ax[1].scatter(y_pr,motor,color = 'blue')
         ax[0].set_title("actual")
         ax[1].set_title('predicted')
         ax[0].set_xlabel('level')
         ax[1].set_xlabel('level')
         plt.show()
```



- We have also tested the accuracy by comparing the predicted data and the actual data as shown in Fig 10

```
In [76]: plt.scatter(x_te[:,-1],y_te,color = 'red')
         plt.scatter(x_te[:,-1],y_pr,color = 'blue')
         plt.title('motor speed vs time')
         plt.xlabel('time')
         plt.ylabel('motor speed')
         plt.show()
```



©IJRASET: All Rights are Reserved

```
In [60]: X_new = X_test[:10]
y_pred = model.predict(X_new)
y_pred=np.round_(y_pred, decimals = 0)
print (y_pred)
print(y_test[:10])

[[3.]
 [3.]
 [2.]
 [2.]
 [3.]
 [2.]
 [1.]
 [3.]
 [1.]
 [2.]]
1593    3
196     3
239     2
2125    2
1587    3
1933    2
1662    1
486     3
790     1
2077    2
Name: level, dtype: int64
```

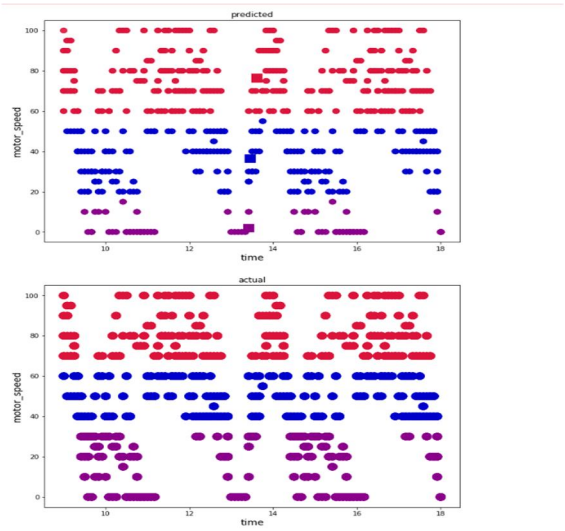


Fig 10: Actual Vs Predicted data value

1) How the Efficiency is Increased

We reduce the parameter in the data by excluding the days and integrating the water level into one column.

	A	B	C	D	E	F	G
1	Time	days	L3	L2	L1	motor_spe	weeks
2	9:00:00 AM	Monday	1	0	0	100	week 1
3	9:00:00 AM	Tuesday	1	0	0	80	week 1
4	9:00:00 AM	Wednesda	1	0	0	90	week 1
5	9:00:00 AM	Thursday	0	1	0	60	week 1
6	9:00:00 AM	Friday	1	0	0	70	week 1
7	9:05:00 AM	Monday	1	0	0	95	week 1
8	9:05:00 AM	Tuesday	1	0	0	80	week 1
9	9:05:00 AM	Wednesda	1	0	0	90	week 1
10	9:05:00 AM	Thursday	0	1	0	50	week 1
11	9:05:00 AM	Friday	1	0	0	70	week 1
12	9:10:00 AM	Monday	1	0	0	95	week 1
13	9:10:00 AM	Tuesday	1	0	0	70	week 1
14	9:10:00 AM	Wednesda	1	0	0	80	week 1
15	9:10:00 AM	Thursday	0	1	0	50	week 1
16	9:10:00 AM	Friday	1	0	0	70	week 1
17	9:15:00 AM	Monday	1	0	0	80	week 1
18	9:15:00 AM	Tuesday	1	0	0	70	week 1
19	9:15:00 AM	Wednesda	1	0	0	75	week 1

Fig 11: Data set with day wise per week

A	B	C
minutes	otor_spee	level
0	100	3
30	30	1
60	10	1
90	100	3
120	50	2
150	80	3
180	100	3
210	40	2
240	0	1
270	70	3
300	50	2
330	70	3
360	30	1
390	0	1

Fig 12: Data set with time and speed

- In the first figure, Fig 11, the time column is taken in a 5 min interval and follows a 5 repetitive entry of each time instance for different days of the week.
- The next columns L1, L2, and L3 indicate the liquid level indicated is by 1
- The 6th column consists of motor speed percentage
- The last column indicates the week

The tank consists of 3 liquid levels denoted as level 1, level 2, and level 3. we have considered 0%-35% motor speed as level 1, this indicates that the liquid is almost filled hence its motor speed is low.
level 2 is in the range of 35%-65% motor speed. this indicates that the tank is partially filled and moderate motor speed is required.
Level 3 is in the range of 65% to 100% motor speed. this indicates that tank is empty and needs the motor to run at a higher speed
In the next figure, we have reduced the level column from 3 to 1 and also have converted the cyclic time dataset to a series dataset to improve the speed as well as the accuracy
We also converted the cyclic working hour into a finite set of minutes which has an interval of 30 min as shown in the figure We also added more data to the dataset

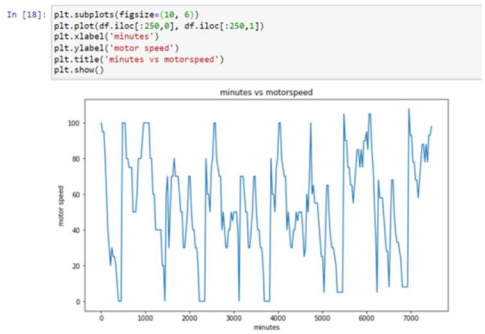


Fig 13: Graph of Minutes Vs Speed of motor.

2) Predicting the level of the tank takes place with three different cases explained below as following

a) First Case: The motor speed is adjusted based on the speed we have set for the particular time and level. the system can also predict the future speed and make its own decision to increase the speed drastically or slowly. we can see from the graph above how the motor speed is varied based on the time

b) Second Case: If the speed is accidentally altered, the system can adjust to the right speed. this is done by predicting the output of the current speed and comparing it with the default speed. In case the next time instance the speed required is less than the current speed or if the level is much higher than the current level it will adjust to an optimal speed

c) Third Case: If the liquid level changes arbitrarily then the system can adjust the motor speed based on the predefined speed and also based on the future level of liquid to set an optimal motor speed



Fig 14: Plot of time, speed and level

The above graph shows the relation between time, motor speed, and level. the time is plotted on the x-axis and the motor speed on the y-axis. the liquid level is denoted by 3 different colors as shown above.

Explanation?

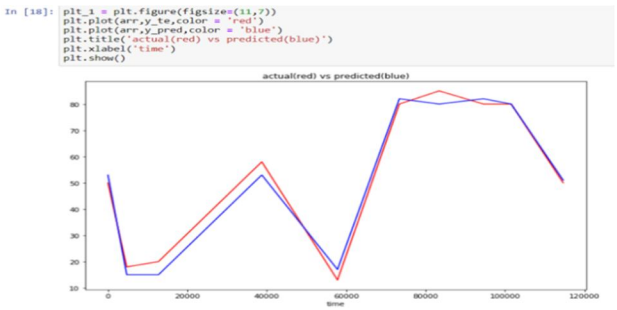


Fig 15: plot of actual vs predicted Speed

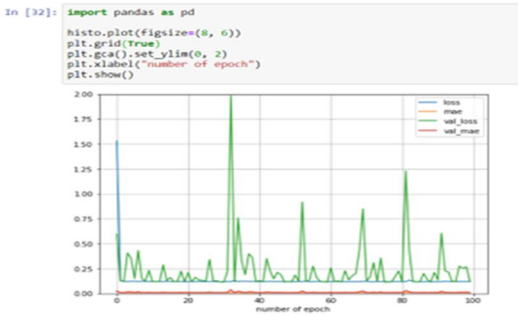
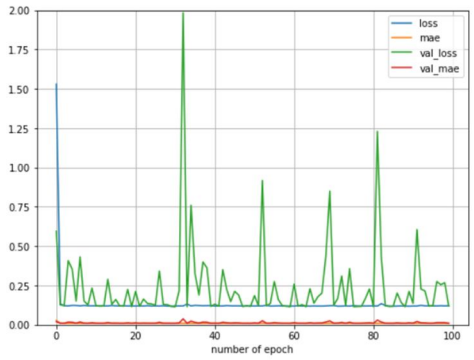


Fig16:trained data with loss and Mae

The above graph shows the relation between actual speed and the predicted speed. the red denotes actual speed and the blue for predicted motor speed. the next figure shows the graph of output of the trained ANN model which consists of loss and mae.
The x-axis is the number of epochs and the y-axis are floats.
During the first few epoch, we will notice that the error is large. this variation is caused due to The cost function being very large as seen in the fig above. during the second iteration, the model is being calculated based on the initial values and the parameter estimates are improved over several steps such that the cost function assumes a minimum value eventually. we notice that there is more fluctuation over every epoch as the resolution is less compared to 5 min and 15 min this causes weight to oscillate and attend an unstable minimum error.

```
In [32]: import pandas as pd

         histo.plot(figsize=(8, 6))
         plt.grid(True)
         plt.gca().set_ylim(0, 2)
         plt.xlabel("number of epoch")
         plt.show()
```




```
In [48]: hi=pd.DataFrame(hist.history)
         h=hi.iloc[99:100,:]
         h
```

Out[48]:

	loss	mae	val_loss	val_mae
99	0.117674	0.009645	0.119378	0.009457

Fig 17 : Plot of no of epoch vs loss

B. Case 2:Interval of 5 Min

In case 1 the motor speed and liquid levels are measured at an interval of 5 min as shown in the image below. this datasets is used to train the random forest model as well as artificial neural networks. the results are shown in the below image. we got accuracy of 0.9822 and mean error is 0.009234 .

	A	B	C
1	minutes	motor_speed	level
2	0	100	3
3	5	95	3
4	10	95	3
5	15	80	3
6	20	60	2
7	25	40	2
8	30	30	1
9	35	20	1
10	40	30	1
11	45	25	1
12	50	25	1
13	55	20	1
14	60	10	1
15	65	0	1
16	70	0	1
17	75	0	1
18	80	100	3
19	85	100	3
20	90	100	3
21	95	80	3
22	100	80	3
23	105	75	3
24	110	75	3
25	115	75	3
26	120	50	2
27	125	50	2

Fig 18: data set for 5 minutes

```
In [28]: import pandas as pd

         pd.DataFrame(h).plot(figsize=(8, 5))
         plt.grid(True)
         plt.gca().set_ylim(0, 1)
         plt.xlabel("number of epoch")
         plt.show()
```

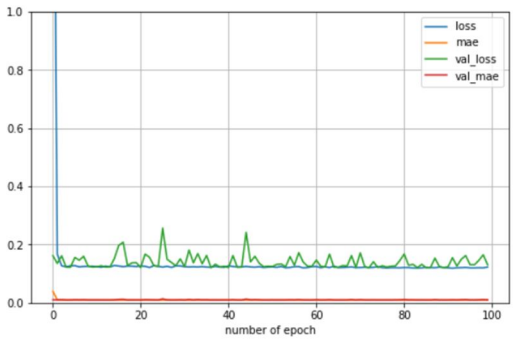


Fig 19: plot of no of epoch vs loss



```
In [14]: y_pred = model.predict(X_new)
         y_pred=np.round_(y_pred, decimals = 0)
         print ("predicted:\n",y_pred)
         print("\n")
         print("actual:\n",y_te)

predicted:
 [[50.]
 [18.]
 [19.]
 [58.]
 [13.]
 [80.]
 [85.]
 [80.]
 [80.]
 [50.]]


actual:
 1606    50
1815    18
2776    20
4928    58
422     13
2225    80
4822    85
2242    80
4768    80
151     50
Name: motor_speed, dtype: int64
```

Fig 20: actual Vs Predicted for 5 min case



```
In [74]: y_pr= model.predict(X_test)
         from sklearn.metrics import r2_score,mean_squared_error
         mse = mean_squared_error(y_test,y_pr)
         r2 = r2_score(y_test,y_pr)
         print(mse)
         print("accuracy")
         print(r2)

120.72647861568126
accuracy
0.9822929294194135
```

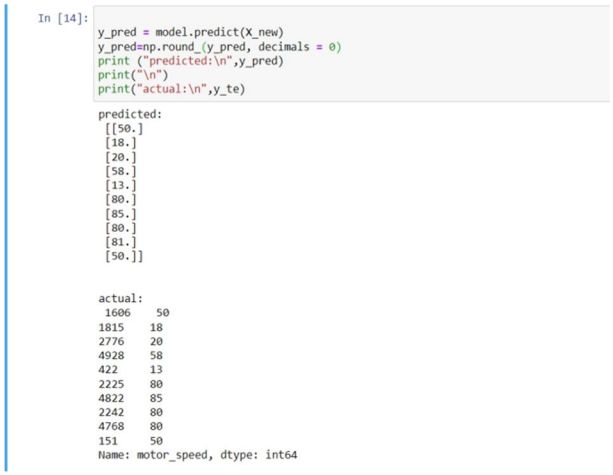
Fig 21: Accuracy and Loss values for 5 min case

C. Case 3: Interval of 15 min

The above dataset is similar to the previous dataset.it was derived from the original dataset by taking an interval of 15 min.In case 1 the motor speed and liquid levels are measured at an interval of 15 min as shown in the image below. this dataset is used to train the random forest model as well as artificial neural networks. the results are shown in the below image. we got are accuracy is 0.97986 And the mean error is 0.009434

A	B	C
minutes	motor_speed	level
0	100	3
15	80	3
30	30	1
45	25	1
60	10	1
75	0	1
90	100	3
105	75	3
120	50	2
135	60	2
150	80	3
165	100	3
180	100	3
195	60	2
210	40	2
225	40	2
240	0	1
255	30	1
270	70	3
285	70	3
300	50	2

Fig 22: Data set sample for 15 min



```
In [14]: y_pred = model.predict(X_new)
         y_pred=np.round_(y_pred, decimals = 0)
         print ("predicted:\n",y_pred)
         print("\n")
         print("actual:\n",y_te)

predicted:
 [[50.]
 [18.]
 [20.]
 [58.]
 [13.]
 [80.]
 [85.]
 [80.]
 [81.]
 [50.]]


actual:
 1606    50
1815    18
2776    20
4928    58
422     13
2225    80
4822    85
2242    80
4768    80
151     50
Name: motor_speed, dtype: int64
```

Fig 23:Actual vs predicted values for 15 min

```
In [32]: import pandas as pd

his.plot(figsize=(8, 6))
plt.grid(True)
plt.gca().set_ylim(0, 2)
plt.show()
```

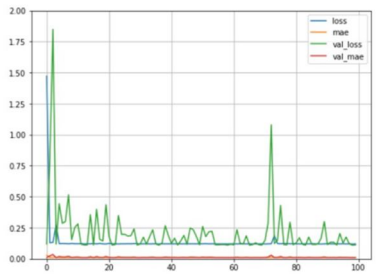


Fig 24: Graph of no Epoch vs loss for 15 min

```
In [48]: hi=pd.DataFrame(hist.history)
         h=hi.iloc[99:100,:]
         h
```

Out[48]:

	loss	mae	val_loss	val_mae
99	0.117674	0.009645	0.119378	0.009457

Fig 25: Loss values

-

D. Case 4: Intervals in random order

In case 1 the motor speed and liquid levels are measured at an interval of 5 min which is randomly organized as shown in the image below.This dataset is used to train the random forest model as well as artificial neural networks. the results are shown in the below image. we got are accuracy is 0.97254 and the mean error is 0.009365

	A	B	C
1	minutes	motor_speed	level
2	23700	20	1
3	38915	96	3
4	25885	56	2
5	29875	28	1
6	2385	16	1
7	9220	78	3
8	23980	30	1
9	17125	103	3
10	30575	83	3
11	8135	40	2
12	15085	73	3
13	2375	66	3
14	28530	86	3
15	36230	26	1
16	38830	96	3
17	21195	0	1
18	8765	85	3
19	37005	0	1
20	4930	46	2
21	20510	43	2
22	35825	23	1
23	31555	40	2
24	29405	65	3
25	34715	5	1
26	22785	93	3
27	28460	0	1

Fig 27: Data set for random Interval

```
In [24]: y_pr= model.predict(X_test)
         from sklearn.metrics import r2_score,mean_squared_error
         mse = mean_squared_error(y_test,y_pr)
         r2 = r2_score(y_test,y_pr)
         print(mse)
         print("accuracy")
         print(r2)

121.89621898819638
accuracy
0.9758877899429489
```

Fig:26 Accuracy for random Interval

```
In [32]: import pandas as pd

histo.plot(figsize=(8, 6))
plt.grid(True)
plt.gca().set_ylim(0, 2)
plt.xlabel("number of epoch")
plt.show()
```

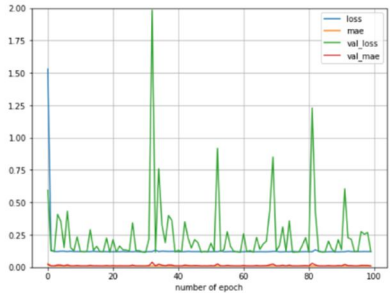


Fig 28: epoch vs loss random Interval

The above datasets is similar to the previous datasets.it was derived from the original datasets by taking an interval of 30 min .The above graph Fig 28 shows the output of the trained ANN model which consists of loss and Mae.

we notice that there is more fluctuation over every epoch as the resolution is more compared to other cases but is jumbled hence during training it might give random parameters are allotted during the learning process causing the model to give the wrong cost function which affects the next epoch to change drastically and causes weight to oscillate and attend an unstable minimum error

```
In [14]: y_pred = model.predict(X_new)
         y_pred=np.round_(y_pred, decimals = 0)
         print ("predicted:\n",y_pred)
         print("\n")
         print("actual:\n",y_te)

predicted:
 [[50.]
 [18.]
 [20.]
 [57.]
 [13.]
 [80.]
 [86.]
 [80.]
 [80.]
 [50.]]


actual:
 1606    50
1815    18
2776    20
4928    58
422     13
2225    80
4822    85
2242    80
4768    80
151     50
Name: motor_speed, dtype: int64
```

Fig 29: Actual Vs Predicted values for random interval

```
In [48]: hi=pd.DataFrame(hist.history)
         h=hi.iloc[99:100,:]
         h
```

Out[48]:

	loss	mae	val_loss	val_mae
99	0.127721	0.009579	0.124374	0.009365

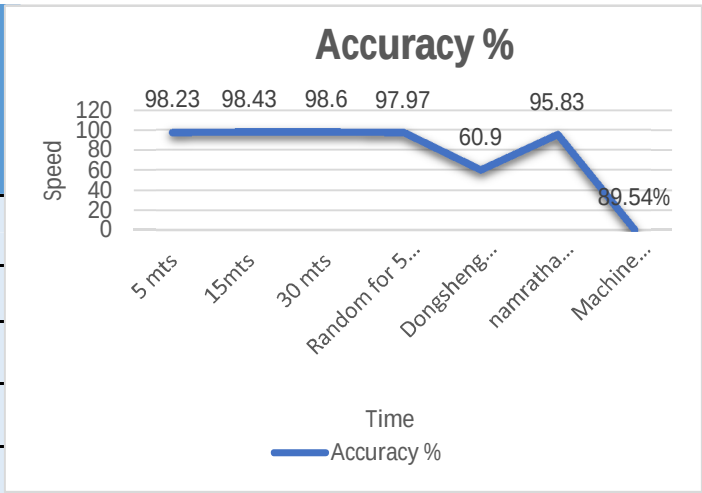
Fig 30: Loss values for random Interval

E. Comparative Analysis of all the cases

1) Comparison analysis of the accuracy of all the cases for the speed Vs the time

Table 1: comparison analysis of accuracy for different cases carried out vs other realted work

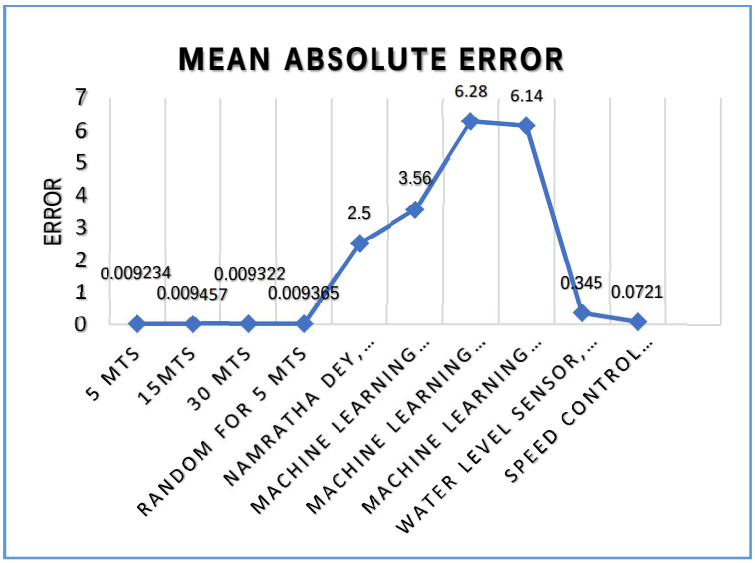
Sl.no	Cases	Accuracy %
1	5 mts	98.23
2	15mts	98.43
3	30 mts	98.6
4	Random for 5 min	97.97
5	Dongsheng wang, RF[1]	60.9
6	namratha Dey, fuzzy logic[2]	95.83
7	Machine learning techniques, FBG [9]	89.54



2) Comparison analysis of mean error recorded for all the cases

Table 2: comparison analysis of MAE for different cases carried out vs other realted work

Sl.no	cases	MAE
1	5 minutes	0.009234
2	15minutes	0.009457
3	30 minutes	0.009322
4	Random for 5 minutes	0.009365
5	Namrata Dey, fuzzy logic[2]	2.5
6	Machine learning techniques, FBG using MM [9]	3.56
7	Machine learning techniques, FBG using SVM [9]	6.28
8	Machine learning techniques, FBG using SVMmin [9]	6.14
9	Water level sensor, fuzzy [11]	0.345
10	Speed control using Machine learning Using NN[10]	0.0721



3) Comparison analysis of the datasets for all the cases

A	B	C
minutes	motor_speed	level
0	100	3
5	95	3
10	95	3
15	80	3
20	60	2
25	40	2
30	30	1
35	20	1
40	30	1
45	25	1
50	25	1
55	20	1
60	10	1
65	0	1
70	0	1
75	0	1
80	100	3
85	100	3
90	100	3
95	80	3
100	80	3
105	75	3
110	75	3

Fig a: Dataset for 5mts

A	B	C
minutes	motor_speed	level
0	100	3
15	80	3
30	30	1
45	25	1
60	10	1
75	0	1
90	100	3
105	75	3
120	50	2
135	60	2
150	80	3
165	100	3
180	100	3
195	60	2
210	40	2
225	40	2
240	0	1
255	30	1
270	70	3
285	70	3

Fig b: Dataset for 15mts

A	B	C
minutes	ıotor_spee	level
0	100	3
30	30	1
60	10	1
90	100	3
120	50	2
150	80	3
180	100	3
210	40	2
240	0	1
270	70	3
300	50	2
330	70	3
360	30	1
390	0	1
420	80	3
450	70	3
480	40	2
510	50	2

Fig c: dataset for 30mts

In the above figure, we can see how all the 3 datasets are co-related to each other. The motor speed and level at 5 min in fig18 is equal to the motor speed and level of 15 min in fig 22. Similarly The motor speed and level at 30 min in fig a are equal to both the motor speed and level of fig b and fig c

4) Comparison of the ANN Model Output of all Cases



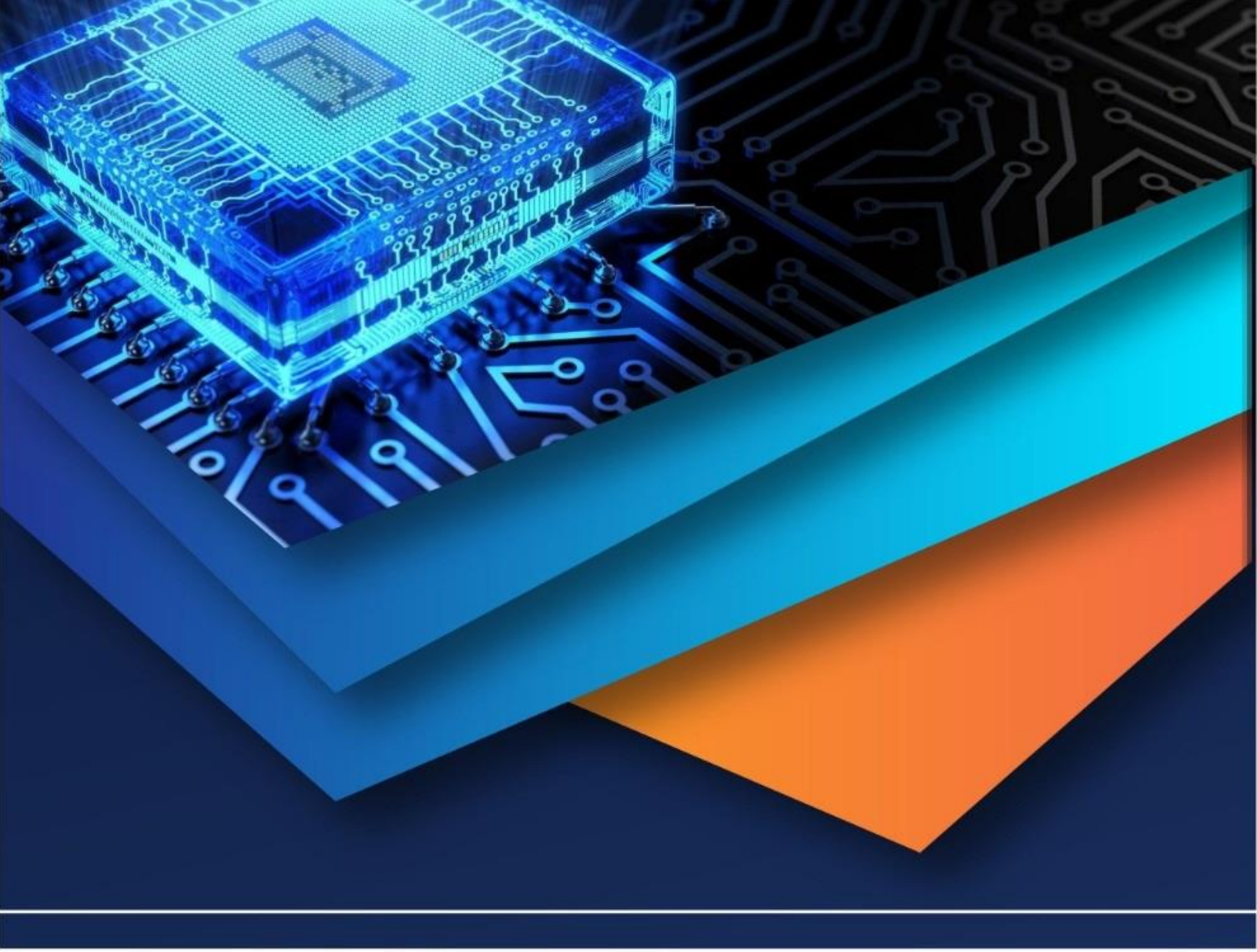
In the first graph, we can see the loss fluctuation per epoch is low compared to the other 2 graphs. This is because the resolution is much higher compared to the other cases. the Mae is almost the same for all 3 cases

VIII. CONCLUSION

Overall its observed that the proposed solution has an overall higher accuracy of 98.6% and above under various testing parameter. This is achieved while maintaining a lower mean absolute error, as low as 0.009234. Considering the precision of the trained machine learning model, its observed from the datasets that motor speed is maintain ensuring efficiency and avoid motor overheating. Comparing the ANN model in various cases also reveals a lower loss fluctuations per epoch as compared to other solutions.

REFERENCES

[1] Water plant optimization control system based on machine learning, Dongsheng Wanga,*, Yan Wanga , Rui Zhoua , Yong Caob , Fuchun Jiangb , Xue Zhangb , Jinghua Lia, doi: 10.5004/dwt.2021.27056, 222 (2021) 168–181.
[2] Design and Implementation of a Water Level Controller using Fuzzy Logic, Namrata Dey Ria Mandal , M Monica Subashini, ISSN : 0975-4024 Vol 5 No 3 Jun-Jul 2013 2277
[3] Automatic Bottle Filling Machine, . B. Kalidasan1, J. Ben Ajai Raja International Research Journal of Engineering and Technology (IRJET) , Volume: 05 Issue: 03 | Mar-2018, p-ISSN: 2395-0072,
[4] Automatic Bottle Filling System Using PLC Based Controller, Md. Liton Ahmed1, Shantonu Kundu, Journal of Advancement in Mechanics Volume 4 , 2019.
[5] Automatic Vessel Filling System Using Raspberry-pi, Pallavi Vivek Shete, Minal V. Raundale, International Engineering Research Journal (IERJ) Volume 2 Issue 3 Page 1449-1451, 2016, ISSN 2395-1621
[6] Industrial Monitoring and Control Using Internet of Things, International Journal for Modern Trends in Science and Technology ,2016
[7] Design and Construct Intelligent Tank“Water Level Sensor, Journal of AL-Qadisiyah for computer science and mathematics Vol.10 No.3 Year 2018ISSN (Print): 2074 – 0204 ISSN (Online): 2521 – 350
[8] Development of an Ultrasonic Sensor Based Water Level Indicator With Pump Switching Technique, June 2018Journal of Electrical Engineering
[9] Machine learning techniques for liquid level estimation using FBG temperature sensor array, Contents lists available at Science Direct Optical Fiber Technology
[10] DC Motor Speed Control Using Machine Learning Algorithm, International Journal of Engineering Research & Technology (IJERT),ISSN: 2278-0181,2018
[11] Design Of Water Level Controller using Fuzzy Logic System , National Institute of Technology Rourkela , 2019
[12] Performance Comparison of Industrial Controller and Fuzzy Logic Controller For Water Level Tank Control System, International Research Journal of Modernization in Engineering Technology and Science (Peer-Reviewed, Open Access, Fully Refereed International Journal),2022
[13] Industrial Monitoring and Control Applications Using IOT, Shrushti Panpaliya,International Journal of Emerging Technologies in Engineering Research (IJETER) Volume 6, Issue 1, January (2018)
[14] Secured Speech Industrial Automation Based on Raspberry Pi and IoT ,Bhagirathi Bai Vittal Rao 1* Hanume Gowda Chandrakanth ,International Journal of Intelligent Engineering and Systems, Vol.11, No.6, 2018 DOI: 10.22266/ijies2018.1231.23
[15] IoT based Machine Learning Automation Algorithm for Controlling the Industrial Loads ,Bhagirathi Bai Vittal Rao1* Hanume Gowda Chandrakanth,International Journal of Intelligent Engineering and Systems, Vol.14, No.4, 2021 DOI: 10.22266/ijies2021.0831.14 .



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)